Sprache & Kognition, 10, Heft 2, S. 61–72

Die konnektionistische Mode[1]

Willem J.M. Levelt
Max-Planck-Institut für Psycholinguistik, Nijmegen

Zusammenfassung. Der vorliegende Aufsatz vergleicht zwei Zugangsweisen zur Modellierung des menschlichen Geistes, nämlich den klassischen symbolischen und den neueren konnektionistischen Ansatz. Insbesondere wird untersucht, inwieweit klassische Fragen wie semantische Kompositionalität und Kohärenz, Lernbarkeit von unendlichen rekursiven Domänen, Regelerwerb, Variablenbindung, usw. adäquat in einem Ansatz behandelt werden, der das klassische Konzept des «stored program» ablehnt. Die Schlußfolgerung ist, daß der Konnektionismus kein geeignetes Beschreibungsniveau für die Behandlung dieser Fragen im Rahmen einer Theorie des Geistes bereitstellt. Am erfolgreichsten wird der Konnektionismus vermutlich beim Modellieren eingeschränkter, nicht-rekursiver Domänen und als Implementierungsmedium sein.

The connectionistic fashion.

Summary. The present essay compares two approaches to the modelling of mind, the classical computational approach and the more recent connectionist one. It is, in particular, considered to what extent classical issues of semantic compositionality and coherence, learnability of infinite recursive domains, rule learning, variable binding, etc. are adequately dealt with in an approach that rejects the classical stored program concept. The conclusion is that connectionism doesn't provide the right level of description for dealing with these issues in the theory of mind. Its best chances are in the modelling of restricted non-recursive domains, and as a provider of implementation media.

Der weitaus interessanteste Aspekt eines halben Jahrhunderts «Informatisierung» ist meiner Meinung nach die Entstehung eines neuen computationalen Modells des menschlichen Geistes. Das Modell hat seinen Ursprung in einer Anzahl mehr oder weniger zusammenhängender Entwicklungen und Entdeckungen.

Vielleicht die wichtigste von ihnen ist Alan Turings (1936) Formalisierung des Begriffs «Berechnung», *computation.* Eine Funktion ist (effektiv) berechenbar, wenn der Wert der Funktion auf rein mechanischem Weg, d.h. ohne Vermittlung einer interpretierenden Instanz, gefunden werden kann.

Das klassische computationale Modell

Schon seit einem halben Jahrhundert besteht die Vorstellung, daß vielleicht jeder gut definierbare Prozeß aus elementaren, mechanisch ausführbaren Berechnungsschritten bestehen könnte, gleichgültig ob es sich nun um eine Beweisführung, eine logische Inferenz, um das Analysieren eines englischen oder holländischen Satzes, das Lösen eines Schachproblems oder das Erfinden einer dreidimensionalen Interpretation für ein zweidimensionales Modell handelt. Die Schritte sind dann von rein syntaktischer Art, d.h. sie beziehen sich ausschließlich auf die Form, nicht auf den Inhalt des eingeführten Ausdrucks.

Turing (1936) konstruierte auch das Modell einer logischen Maschine, die solche Berechnungen ausführen kann. Sie ist eine Kombination aus einem endlichen Automaten und einem unendlich langen Band. Auf einem bestimmten Abschnitt des Bandes ist eine begrenzte Anzahl elementarer Operationen gespeichert, die der Automat ausführen kann (das entspricht dem Programm); der Rest des Bandes ist für die Eingabe, die Speicherung von Zwischenresultaten und die Ausgabe bestimmt. Programm und Daten sind von derselben Art, werden aber bei der Ausführung einer Berechnung streng getrennt gehalten. Schließlich hat Turing den Beweis erbracht, daß *universale Turing-Maschinen* bestehen, d.h. Maschinen, die die Tätigkeit jeder anderen Turingmaschine simulieren können. Das bedeutet, daß

1 Dieser Beitrag erscheint in leicht modifizierter Form in C. Brown, P. Hagoort & Th. Meyering (Eds.): Vensters op de geest. Utrecht: Grafiet, 1989. Die Übersetzung wurde von Carola Engelkamp angefertigt. Ich verdanke Antje Meyer eine Menge kritischer Bemerkungen zum deutschen Text.

Turings Begriff von Berechenbarkeit, von einem effektiven Verfahren, von absolut allgemeiner Art ist.

Es muß bemerkt werden, daß dieser erste Schritt auf dem Weg zur Informatisierung gemacht wurde, bevor es Computer gab. Turings Definition ist vollkommen unabhängig von der Durchführung, die man für das mechanische Verfahren wählt. Es spielt keine Rolle, ob es sich um eine VAX, um ein menschliches Gehirn oder um ein konnektionistisches Netzwerk handelt. Entscheidend für die Berechnung ist das *stored program:* Trennung einer begrenzten Anzahl syntaktischer Operationen und einer unbegrenzten Menge von Daten, auf die die Operationen zurückgreifen können; Semantik und Implementierung sind irrelevant. Solange dieses Prinzip in der virtuellen Maschine verwirklicht ist, d.h. auf der Ebene der logischen Struktur von Daten und Operationen, ist alles in Ordnung.

Eine zweite wichtige Wurzel für unser heutiges *computational model of the mind* liegt in der formalen Logik von Frege, Whitehead and Russell. Hierbei geht es um die Frage: Wie können rein syntaktische Verfahren, die sich nicht um die Bedeutung der Symbole kümmern, auf die sie zurückgreifen, doch semantisch kohärente Resultate produzieren? Wenn man eine Turingmaschine mit wahren Ausdrücken füttert, wie kann man dann dafür sorgen, daß auch wieder wahre Ausdrücke herauskommen? In der formalen Logik wird das Problem dadurch gelöst, daß man die syntaktische Struktur von Formeln oder Ausdrücken systematisch mit der semantischen Interpretation dieser Ausdrücke zusammenhängen läßt.

Wie das funktioniert kann an einem einfachen Beispiel gezeigt werden. Wir wissen alle, daß aus der Behauptung: «Hans fährt Fahrrad, und Peter läuft», «Peter läuft» abgeleitet werden kann. Wenn die erste Aussage wahr ist, ist die zweite es auch. Um dieses Ergebnis zu garantieren, kommt man nun überein, daß jede Aussage, die die syntaktische Struktur «S_1 und S_2» hat, semantisch sowohl S_1 als auch S_2 beinhaltet. Umgekehrt muß jedes der konstituierenden Teile wahr sein, um die Wahrheit der gesamten Aussage zu gewährleisten. Dies ist ein Beispiel für Freges *Kompositionalitätsprinzip:* Die semantische Interpretation (Wahrheit, etc.) einer komplexen Aussage kann aus den semantischen Interpretationen (Wahrheit, etc.) ihrer Konstituenten und den syntaktischen Beziehungen zwischen den Konstituenten abgeleitet werden. Um semantische Kohärenz zu gewährleisten, muß also die formale Sprache, in der man die Aussage schreibt, einen solchen systematischen Zusammenhang zwischen Syntax und semantischer Interpretation aufweisen. Das gilt für alle Computersprachen und nicht zufällig auch in einem sehr hohen Ausmaß für natürliche Sprachen. Es sollte uns sehr wundern, wenn eine Sprache existiert, in der es systematisch der Fall ist, daß aus «Hans fährt Fahrrad und Peter läuft» → «Peter fährt Fahrrad» abgeleitet werden kann. Ausnahmen bestätigen die Regel. Aus «Hans sitzt in Sack und Asche» kann nicht «Hans sitzt in Asche» abgeleitet werden. Ein Idiom ist gerade darum ein Idiom, weil es nicht dem Kompositionalitätsprinzip entspricht.

Das computationale Modell des menschlichen Geistes, das sich im letzten halben Jahrhundert entwickelt hat, beruht auf diesem Prinzip. Auch die *language of thought,* die Sprache (oder Sprachen), in der sich unsere mentalen Operationen abspielen, zeigt diese systematische Beziehung zwischen syntaktischer Konstituentenstruktur und semantischer Interpretation. Auf diese Weise können rein mechanische, aber doch strukturabhängige Operationen semantisch kohärente Ergebnisse liefern.

Eine seit Gödel und Turing mögliche, aber erst seit den 50er Jahren angewandte Form semantischer Interpretation ist es, auf gespeicherte Algorithmen oder Programme als *Daten* zu verweisen. Die Operation wird dann nicht ausgeführt, sondern zitiert. Die zitierte Aussage kann eventuell verändert werden, so daß eine neue Operation entsteht. Auf diese Weise kann das Programm sich selbst modifizieren oder ein neues Programm kreieren. Diese Möglichkeit der Selbstorganisation innerhalb der klassischen Architektur ist ein zentraler Punkt in SOAR, einer Lerntheorie, die von Laird, Rosenbloom und Newell (1986) entwickelt wurde.

Allmählich ist man zu der Überzeugung gelangt, daß solche semantisch interpretierten physischen Symbolsysteme notwendig und hinreichend sind, um jedes intelligente Verhalten zu modellieren. Die Arbeit von Newell, Simon et al. beruht auf dieser Überzeugung. Ihre Arbeit wird völlig zu Unrecht noch stets als Modell für das

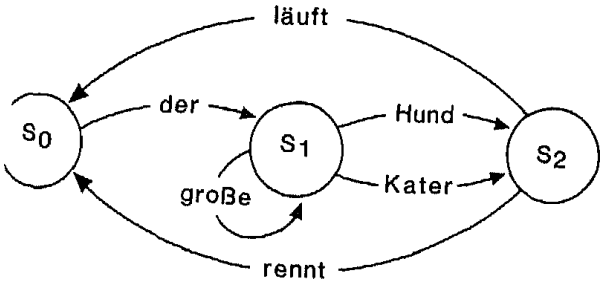
langsame, bewußte, serielle menschliche Denken angesehen, wie es beim Schach oder beim Beweisen einer Theorie stattfindet.

Das bringt uns zur dritten Wurzel des computationalen Modells des menschlichen Geistes, den Entwicklungen in der Sprachwissenschaft. In den 50er Jahren machte Chomsky (1957) auf die unbegrenzte *Produktivität* natürlicher Sprachen aufmerksam. Für jeden wohlgeformten Satz gibt es einen längeren, der auch wohlgeformt ist, d.h. die Menge wohlgeformter Sätze ist unbegrenzt. Es ist die Aufgabe der Sprachwissenschaft, diese Produktivität mit Hilfe eines endlichen rekursiven Mechanismus zu beschreiben. Chomsky konnte damals beweisen, daß dies mit einem rekursiven assoziativen Mechanismus wie dem Markov-Prozeß, oder allgemeiner mit einem endlichen Automaten, unmöglich ist (siehe Levelt, 1973, für Definitionen dieser Begriffe und für Chomskys Beweis. Abbildung 1 gibt ein Beispiel für einen solchen endlichen Automaten).

Ein endlicher Automat kann sich in einer begrenzten Anzahl von Zuständen (Si) befinden, unter denen ein Anfangs- und ein Endzustand sein müssen, und er hat ein begrenztes Vokabular V an Ein- und Ausgabesymbolen. Der Automat kann seinen Zustand verändern, wenn ein dazu geeignetes Symbol eingegeben (oder produziert) wird.

Übergangsregeln geben an, welcher neue Zustand erreicht wird, wenn in einem bestimmten Zustand ein neues Symbol eingegeben wird. Ein Übergangsdiagramm verdeutlicht den Vorgang.

Als Beispiel sieht man hier ein Übergangsdiagramm für einen endlichen Automaten mit drei Zuständen: S0 (Anfangs- und Endzustand), S1 und S2. Das Vokabular besteht aus sechs Wörtern: der, große, Hund, Kater, läuft, rennt. Die Wörter stehen bei den Zustandsübergängen, die sie ermöglichen.



Im Zustand S0 beginnend und endend, kann dieser Automat Sätze wie «der Hund läuft», «der große Kater rennt», «der große, große Hund rennt» usw. akzeptieren (oder produzieren). Wenn man den Zustandsübergängen Wahrscheinlichkeiten zuordnet, dann erhält man einen Markov-Prozeß. Die Summe der Übergangswahrscheinlichkeiten, von einem Zustand ausgehend, ist 1. Mit dem Markov-Prozeß kann man voraussagen, wie wahrscheinlich die «Sätze» sind, die ein endlicher Automat produziert.

Mengen, die durch einen endlichen Automaten produziert oder aktiviert werden, heißen reguläre Mengen.

Abbildung 1: Endliche Automaten

Es geht hier darum, daß man die Sätze in einer natürlichen Sprache nicht dadurch konstruieren kann, daß man jedes folgende Wort allein und ausschließlich aufgrund des zuletzt produzierten Wortes wählt, also durch Ausführen einer Reihe lokaler Assoziationen. Die Syntax natürlicher Sprachen weist rekursive, *hierarchische* Eigenschaften auf, die nur von einem wesentlich komplexeren Automaten realisiert werden können. Es handelt sich hier also um eine empirische Eigenschaft natürlicher Sprachen, um einen bestimmten Typ von Rekursivität, den wir Menschen scheinbar in uns haben, ohne uns davon bewußt zu sein. Völlig unbewußt verbinden wir Variablen in Abhängigkeit von der rekursiven, hierarchischen Struktur linguistischer Aussagen. Ein Beispiel hierfür gibt Abbildung 2, wo das Wort «sich» abhängig von der hierarchischen Struktur des Satzes jeweils durch ein anderes Wort in den Satz eingebunden wird.

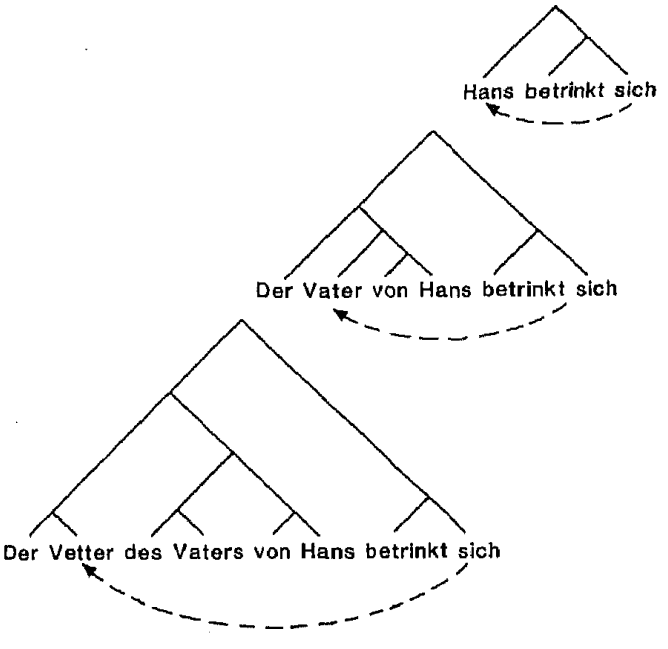


Abbildung 2: Rekursive Syntax und hierarchische Struktur

Mit diesen Entdeckungen wird das behavioristische Modell des menschlichen Sprachverhaltens, das ja explizit auf einem assoziativen Markov-artigen Mechanismus basierte, zu Recht als völlig unzureichend beiseite gelegt. Seitdem hat es eine wahre Flut von Untersuchungen über die abstrakten linguistischen Regeln gegeben, die

jeder von uns andauernd und mit verblüffend wenigen Fehlern anwendet. Immer noch werden täglich neue Regeln und Prinzipien entdeckt. Diese Regeln werden keineswegs bewußt konstruiert, gelernt oder verwendet. Die Prozesse des Sprechens und Verstehens verlaufen sowieso viel zu schnell, als daß man die Regeln bewußt anwenden könnte. Eine der artspezifischen Fähigkeiten des Menschen, die Sprache, scheint ein produktives rekursives System zu sein, das nicht auf assoziativen, sondern auf hierarchischen syntaktischen Operationen beruht und größtenteils unbewußt und in paralleler Verarbeitung abläuft. Solche Fähigkeiten konnte man selbst bei den Schimpansen, unseren nächsten Nachbarn im Tierreich, nicht nachweisen.

Eine vierte wichtige Entwicklung kann ich hier nur andeuten. Es ist die Theorie der propositionalen Einstellungen *(propositional attitudes).* Menschen handeln auf Grund dessen, was sie wissen, denken, meinen, hoffen, wünschen, beabsichtigen usw. Dies sind Einstellungen, die wir gegenüber Propositionen einnehmen, und die als Ursache für unser Verhalten angesehen werden können. Seit einigen Jahren werden in der Formalisierung, d.h. bei der mechanischen Berechnung von propositionalen Einstellungen, große Fortschritte gemacht (Levelt, 1988). Hiermit wird eine formale Basis für eine Theorie des rationalen Verhaltens geschaffen.

Ich habe nun einige wesentliche Eigenschaften des computationalen Modells des menschlichen Geistes genannt, wie es sich im letzten halben Jahrhundert Hand in Hand mit der Informatisierung entwickelt hat. Kurz zusammengefaßt geht es um ein mechanisches oder physisches Modell einer sehr bestimmten Architektur. Geistige Operationen sind von rein syntaktischer Art, d.h. sie sind durch die Form, nicht durch den Inhalt von Aussagen bestimmt. Semantische Kohärenz wird durch Freges Prinzip der Kompositionalität garantiert. Man muß streng unterscheiden zwischen endlichen Mengen mentaler Operationen und im Prinzip unendlichen Mengen von Aussagen oder Daten, auf die die Operationen wirken können, oder die durch die Operationen produziert werden können. Das bezeichnet man als Produktivitätsprinzip. Eine Beschränkung oder Erweiterung des Arbeitsgedächtnisses läßt das Programm unbeeinflußt, hat aber vorhersagbare Folgen für die Art der erzeugten Aussagen und die Art des Verhaltens. Mentale Operationen oder Produktionsregeln sind wesentlich komplexer als Assoziationen. Sie sind in dem Sinne strukturabhängig, daß sie an der hierarchischen Struktur von Aussagen ansetzen. Schließlich werden nun auch die Einstellungen des Menschen – das System der propositionalen Einstellungen, auf denen menschliche Entscheidungen basieren – in das computationale Modell einbezogen. Symbolsysteme mit diesen Eigenschaften sind virtuelle Maschinen. Es ist die Überzeugung gewachsen, daß logische Strukturen eine Erklärungsebene *sui generis* bilden, unabhängig davon, wie solche virtuellen Maschinen implementiert werden. Damit ist nicht gesagt, daß die Implementierung irrelevant ist, sondern nur, daß sie nicht die geeignete Erklärungsebene für mentale Prozesse ist (genausowenig wie die atomare Ebene das geeignete Erklärungsniveau für geologische Prozesse oder die neurologische Ebene für ökonomische Prozesse ist).

Das konnektionistische Modell

Als ich vor 15 Jahren noch Psychologiestudenten im ersten Semester unterrichtete, beklagten sich einige Studenten über die Tatsache, daß Professor X ihnen fortlaufend erzähle, wie unsinning der Behaviorismus doch sei, daß sie aber überhaupt nicht wüßten, was genau Behaviorismus bedeute. Ähnliches geschieht wahrscheinlich, wenn ich hier am Konnektionismus Kritik übe. Nicht jeder weiß, was Konnektionismus ist, und ein noch geringerer Teil ist bereit, sich mit ihm auseinanderzusetzen. Aber im Gegensatz zum aussterbenden Behaviorismus in den siebziger Jahren ist der Konnektionismus von heute jung und quicklebendig und hat alle Merkmale einer Modeerscheinung mit enormer Anziehungskraft.

Vor allem in Amerika und England nimmt die neue «Schule» revolutionäre Ausmaße an. Zusammenkünfte der «Cognitive Science Society» z.B. werden beherrscht von konnektionistischen Beiträgen und Diskussionen. Es wimmelt nur so von konnektionistischen Workshops und Konferenzen. In vielen psychologischen Fakultäten machen Studenten nichts anderes mehr als konnektionistische Modelle zu bauen, und die Zeitschriften sind voll davon, was konnektionistische

Netzwerke schon wieder alles gelernt haben.

In Holland und Deutschland geht es zum Glück noch nicht so extrem zu. Die konnektionistische Forschung, die hier stattfindet, ist von guter Qualität, und die Rhetorik hält sich in vernünftigen Grenzen. Eine übersichtliche und ausgeglichene Einführung in den Konnektionismus findet man in Goebel (1990). Die Entwicklungen im Ausland zwingen allerdings zu Wachsamkeit, daher dieser Artikel.

Das konnektionistische Modell der menschlichen Kognition unterscheidet sich nahezu in allen wesentlichen Punkten von dem gerade beschriebenen computationalen Modell. Die geeignete Beschreibungsebene der menschlichen Kognition ist nach Meinung der Konnektionisten nicht die Ebene der virtuellen symbolmanipulierenden Maschine, sondern die subsymbolische Ebene. Die subsymbolische Ebene besteht aus Knoten, die miteinander verknüpft sind (siehe Abb. 3).

Die Knoten sind Einheiten, die einen ganz einfachen Prozeß ausführen. Zu jedem Zeitpunkt befindet sich jeder Knoten in einem bestimmten Aktivitätszustand (a). Der Aktivitätszustand des gesamten Systems zum Zeitpunkt t ist also der Zustandsvektor (a_t). Über diese Verbindungen kann jeder Knoten Signale aussenden und empfangen. Das Signal, das ein Knoten zum Zeitpunkt t aussendet, ist eine Funktion des Aktivitätszustandes. Diese Funktion heißt die Ausgabefunktion des Knotens. Das Ausgangssignal (o_t) wird über die Verbindungen zu den anderen Knoten weitergeleitet. Jede Verbindung hat ihre eigene Stärke oder ihr eigenes Gewicht (w). Dieses ist eine positive oder negative reelle Zahl. Was die Verbindung durchgibt, ist das Produkt aus Gewicht und Ausgabesignal ($w \times o$). Dieses bildet das Eingabesignal für den empfangenden Knoten. Ein Knoten kann zu einem bestimmten Zeitpunkt t über verschiedene Verbindungen Signale empfangen. Die Inputsignale werden gemäß einer bestimmten Eingabefunktion addiert. Meist wird hierfür eine quasi-lineare Funktion gewählt: Alle Signale werden linear addiert, und die Summe wird nicht-linear transformiert. Dies soll verhindern, daß die Aktivierung eines Knotens beliebig groß werden kann.

Das Verhalten eines solchen konnektionistischen Netzwerkes, d.h. die Veränderung des Zustandsvektors über die Zeit, kann im Prinzip durch eine Reihe von Differentialgleichungen beschrieben werden. Meist benutzt man allerdings eine diskrete Zeitskala, weil das für Computersimulationen einfacher ist.

Eine bestimmte Teilmenge von Knoten wird als Eingabeknoten *(input nodes)* definiert, eine andere Teilmenge als Ausgabeknoten *(output nodes)*. Alle anderen Knoten nennt man verborgene Knoten *(hidden nodes)*. Das Netzwerk wird nun dadurch stimuliert, daß die Eingabeknoten entsprechend einer bestimmten statistischen Verteilung aktiviert werden. Dieses konstante statistische Aktivitätsmuster wird nun so lange beibehalten, bis die Ausgabeknoten ebenfalls eine konstante Aktivierungsverteilung aufweisen. Dies ist die Antwort des Systems. Das Netzwerk setzt also einen statistischen Eingabevektor in einen statistischen Ausgabevektor um. Die Beziehung, die sich zwischen Eingabe- und Ausgabevektor bildet, wird hauptsächlich durch die Stärke der Verbindungen im Netzwerk bestimmt.

Es gibt global gesehen zwei Möglichkeiten, mit einem solchen Netzwerk kognitive Prozesse zu

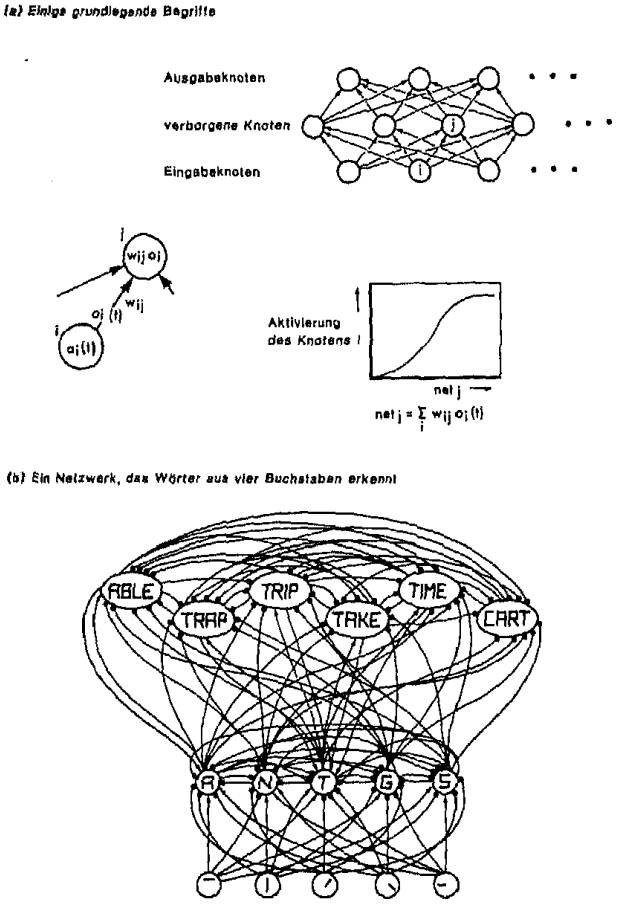


Abbildung 3: Konnektionistische Netzwerke

modellieren. Bei der ältesten, aber uninteressantesten, programmiert man selbst alle Verbindungen, d.h. man ordnet den Verbindungen mit der Hand Werte zu. Auf diese Weise ist es z.B. relativ gut gelungen, das System Wörter aus vier Buchstaben erkennen zu lassen (wie ABLE, TRAP, TIME). Es gibt dann Eingabeknoten für jeden schrägen oder geraden Strich in jedem der vier Buchstaben (die sogenannten Buchstabenmerkmale). Die Ausgabeknoten sind die möglichen Wörter. Die Kunst besteht nun darin, den Verbindungen in dem Netzwerk solche Gewichte zuzuordnen, daß bei einem bestimmten Eingabemuster von Buchstabenmerkmalen der richtige Ausgabeknoten aktiviert wird. So etwas ist zwar möglich (McClelland & Rumelhart, 1981), aber besonders aufregend ist es nicht, wenn man an die enorme Anzahl freier Parameter und Verbindungsstärken denkt, die einem zur Verfügung stehen (siehe Abb. 3).

Die zweite Methode ist interessanter. Man läßt das Netzwerk selbst alles lernen. Man bietet ihm nun Paare von Eingabemustern und gewünschten Ausgabemustern dar. Beim ersten Durchlauf wird das tatsächliche Ausgabemuster nicht dem gewünschten Ausgabemuster entsprechen. Für jeden Ausgabeknoten wird der Unterschied berechnet, und auf Grund dieser Differenz werden mit Hilfe einer Technik, die *backpropagation* genannt wird, die Gewichte der Verbindungen automatisch ein klein wenig modifiziert. Dadurch, daß dieses Verfahren nun fortlaufend mit einer bestimmten Menge von Eingabe- und Ausgabepaaren wiederholt wird, kann es passieren, daß so ein Netzwerk die richtige Assoziation lernt. Es kann dann manchmal auch bei neuen Stimuli – die es vorher noch nicht «gesehen» hat – die richtige Antwort generieren. Das Netzwerk hat dann scheinbar eine Regel oder eine Menge von Regeln gelernt. Ein Beispiel dafür ist ein Netzwerk, das lernt, die Vergangenheitsform englischer Verben zu produzieren: *walk – walked, bite – bit* etc. Das Netzwerk bildet dann auch oft für Verben, die es vorher noch nicht gesehen hat, die richtigen Vergangenheitsformen.

Von diesem Beispiel ausgehend, kann nur der Hauptgedanke der Konnektionisten erläutert werden. Es gibt keine expliziten Regeln im Netzwerk und keine symbolische Repräsentation, sondern nur ein Muster von Assoziationen. Trotzdem verhält sich das Netzwerk mehr oder weniger entsprechend eines Systems von (in diesem Fall linguistischen) Regeln. Es ist allerdings nicht so, daß die Regeln intern gespeichert sind und somit das Verhalten verursachen, wie das beim computationalen Modell der Fall ist. Die Regeln sind vielmehr *emergent properties,* Epiphänomene. Die wirklichen Verhaltensdeterminanten befinden sich auf der subsymbolischen Ebene; es sind die Verbindungsmuster im Netzwerk. Nach den Konnektionisten ist dies die einzig angemessene Erklärungsebene für mentale Prozesse.

Obwohl der Konnektionismus alles andere als neu ist – er hat eine lange Vorgeschichte in der Theorie neuraler Netzwerke von McCulloch & Pitts (1943), Hebb (1949), Rosenblatt (1962) und Minsky & Papert (1969) –, lassen sich die enormen Aktivitäten von heute nur dadurch erklären, daß große bis sehr große Computer zur Verfügung stehen. Erst jetzt ist es innerhalb absehbarer Zeit möglich zu testen, ob ein großes assoziatives Netzwerk einen nicht völlig trivialen Wissensbestand erwerben kann. Das normale konnektionistische Vorgehen besteht dann auch darin, die konnektionistische Methode an allen möglichen Wissensgebieten auszuprobieren, wie etwa dem Lernen von visuellen Mustern, dem Lernen von syntaktischen und phonologischen Regeln, dem Erkennen von Wörtern, dem Lernen von Addition und vielem mehr. Man definiert das zu lernende Gebiet, stellt einen Lehr- oder Darbietungsplan auf, gibt alles am Abend in einen großen Computer ein und schaut am nächsten Morgen, wie gut das Netzwerk damit umgehen kann, und wieviele Darbietungen nötig waren, um das Wissen zu erlernen. Wenn das alles einigermaßen gelungen ist, schließt man, daß das Gebiet durch ein subsymbolisches assoziatives Netzwerk erlernbar ist, und deshalb für diesen Bereich für eine klassische symbolische Berechnungsarchitektur kein Bedarf mehr besteht.

Konnektionistische Rhetorik

Es ist natürlich ratsam, die wesentlichen theoretischen Fragen gut von der Rhetorik getrennt zu halten, aber es kann nicht schaden, einen Augenblick bei der Rhetorik zu verweilen. Es ist nämlich nicht immer einfach, in den Schriften der

Konnektionisten Tatsachen und Phantasie auseinanderzuhalten. Der Schein hält oft nicht, was er verspricht. Die Verpackung des Konnektionismus besteht nicht selten aus einer übertriebenen Darstellung des eigenen Könnens und einer abwertenden Darstellung der Leistungen der klassischen Architektur. Beides ist häufig unbegründet oder sogar nachweisbar falsch. Konnektionismus sollte eine *«theory from which the multiplicity of conceptual theories can be seen to emerge»* sein (Hunter, 1988); *«It is likely that connectionist models will offer the most significant progress of the past several millenia on the mind/body problem»* (Smolensky, 1988). Konnektionistische Netzwerke sollten *«effectively Turing machines»* sein (Elman, 1989), eine Behauptung, die nicht aus den Prämissen folgt (siehe unten). Die Aufhebung der Trennung zwischen Programm und Daten wird als theoretischer Durchbruch dargestellt. (Das ist sie auch, aber in einem negativen Sinn). Man weist mit Enthusiasmus auf den chaotischen Charakter von konnektionistischen Systemen hin, der sie für eine Selbst-Organisation so geeignet machen soll. Konnektionistische Modelle sollen uns aufs Neue hoffen lassen, daß wir Erscheinungen wie Intuition, *common sense,* Kreativität, Bewußtsein und Selbstbewußtsein verstehen können. Kurzum, viel Glaube und Hoffnung. Und auch viel Haß. Die klassische Architektur und die Phänomene, die sie zu erklären beabsichtigt, werden höhnisch zur Seite geschoben: *«recursive progressing (is not) of the essence of human computation»* (Rumelhart & McClelland, 1986, p. 119), *«there is no induction problem. The child need not figure out what the rules are, not even that there are rules»* (Rumelhart & McClelland, 1986, p. 267). Es wird wiederholt und zu Unrecht behauptet, daß symbolverarbeitende Modelle nur Symbole und Regeln zulassen, die bewußt zugänglich sind, daß sie nur sequentielle und keine parallele Verarbeitung zulassen, daß sie nicht imstande sind, abweichendes Verhalten, Unregelmäßigkeiten und Fehler vorauszusagen oder zu erklären usw.

Man kann ohne Rhetorik keine Wissenschaft betreiben. Aber es gibt wohl Anlaß zur Sorge, wenn wir, so wie es heutzutage in großem Umfang geschieht, unsere Studenten mit Illusionen vollstopfen, wenn wir sie wie Analphabeten erziehen, die keine Ahnung davon haben, was auf dem Gebiet der computationalen Theorien im letzten halben Jahrhundert für fundamentale Erkenntnisse erworben wurden, und wenn wir ihnen die Vorstellung vermitteln, daß eine empirische Untersuchung darin besteht zu testen, ob ein Modell etwas lernen kann, statt festzustellen, ob Menschen etwas lernen können. Es gibt keine empirische (experimentelle) konnektionistische Psychologie.

Lassen wir die Rhetorik das sein, was sie ist, und analysieren wir einige theoretische und empirische Probleme, die sich dem konnektionistischen Modell des menschlichen Geistes stellen.

Einige theoretische und empirische Probleme

Der Konnektionismus wirft jede essentielle Errungenschaft von mehr als einem halben Jahrhundert computationaler Theorie über Bord. Die erste besteht darin, daß mentale Prozesse als syntaktische Operationen an symbolischen Repräsentationen aufgefaßt werden können. Die zweite ist die Erkenntnis, daß die semantische Kohärenz einer syntaktischen Maschine auf Freges Kompositionalitätsprinzip beruhen muß, d.h. auf der hierarchischen Konstituentenstruktur symbolischer Ausdrücke. Die dritte Errungenschaft ist die Unterscheidung zwischen einem endlichen gespeicherten Programm und einer im Prinzip unendlichen Menge von Daten (das *stored program*) und damit eng zusammenhängend zwischen einer begrenzten computationalen Struktur und einem schier unbegrenzten Wissensbestand. Und damit verschwindet auch eine vierte Errungenschaft, nämlich die Erkenntnis, daß die menschliche Kognition mit begrenzten computationalen Mitteln unbegrenzte Wissensbestände hervorbringen oder interpretieren kann (das Produktivitätsprinzip). Schlußendlich wird dann noch die Annahme verworfen, daß echte Erklärungen auf symbolischer Ebene möglich sind, vor allem durch Bezug auf propositionale Einstellungen wie Wünschen, Meinungen, Zielsetzungen und Überzeugungen. Dies sollen vielmehr *emerging properties* sein, d.h. Epiphänomene. Die Trennung der Geister ist hier nahezu vollkommen. Trotzdem will ich eine Reihe von Problemen skizzieren, die unwiderruflich auftauchen, wenn man diese drastischen Schritte vollzieht.

Ein erstes Problem betrifft die Erlernbarkeit von Wissensbeständen. Wir haben gesehen, daß Konnektionisten sehr an der Erlernbarkeit von allerlei Wissensbeständen interessiert sind. Kann man einem Netzwerk eine bestimmte Menge von Wörtern, visuellen Objekten, Sätzen oder logischen Operationen beibringen? Die Frage, ob etwas erlernbar ist, wird stets dadurch beantwortet, daß man es einfach auf dem Computer ausprobiert. Dies ist ein äußerst unprinzipieller Weg (Levelt, 1990a). Man darf nämlich aus der Erlernbarkeit eines bestimmten Wissensbestandes nicht schließen, daß ein größerer Wissensbestand derselben Art auch erlernbar ist. Angenommen, ein konnektionistisches Modell kann eine Sprache mit Sätze vom Typ «Wenn Hans sagt, daß es regnet, schwindelt er» lernen. Dann gibt es noch keine Garantie dafür, daß das Modell mit Sätzen wie «Wenn Hans sagt, daß Peter sagt, daß es regnet, schwindelt er» zurechtkommt. Aber vielleicht kann ein viel größeres Netzwerk das auch noch. Leider gibt es dann wieder keine Garantie dafür, daß das Netzwerk mit Sätzen wie «Wenn Hans sagt, daß Peter sagt, daß Claus sagt, daß es regnet, dann schwindelt er», usw. umgehen kann. Kurzum, wir können auf diese Art niemals erfahren, ob ein Netzwerk eine Sprache erlernen kann, die unbegrenzte Rekursion dieser Art zuläßt. (So eine Sprache heißt «kontextfreie Sprache».)

Computersimulation eines Modells ist institutionalisierte Faulheit. Was wirklich notwendig ist, ist einen Beweis zu erbringen, ob etwas erlernbar ist oder nicht, und wenn es nicht erlernbar ist, wo sich dann die Asymptote für Erlernbarkeit befindet (z.B. bei zwei-, drei-, oder vierfacher Rekursion). In der klassischen Tradition und interessanterweise auch bei den Vorläufern des Konnektionismus, McCulloch, Pitts, Rosenblatt, Minsky und Papert, war das auch die normale Vorgehensweise. Man hat immer bewiesen, ob eine Menge lernbar ist oder nicht.

In der klassischen Tradition ist viel über die Erlernbarkeit von allerlei Wissensbereichen wie die von kontextfreien und anderen Sprachen in Erfahrung gebracht worden. Innerhalb des heutigen Konnektionismus mangelt es dagegen vollkommen an Erlernbarkeitsbeweisen. Es gibt also keine Basis für Generalisierungen. Jede Behauptung, daß ein konnektionistisches Modell X erlernen kann, wobei X eine nicht-reguläre Menge ist, ist vorläufig unbewiesen. Hiermit eng verknüpft ist der Mangel an Verstehbarkeit eines Ergebnisses. Wenn ein konnektionistisches Modell etwas gelernt hat, kann man sagen: «Es hat X gelernt» und «Es hat solange dafür gebraucht», aber man weiß dann noch lange nicht, warum es X lernen konnte, oder warum es Y nicht lernen kann. Wir entdecken auf diese Weise keine Erklärungsprinzipien. Ein konnektionistisches Modell ist *mutatis mutandis* genau so praktisch wie ein Stadtplan im Maßstab 1:1.

Das Fehlen von Erlernbarkeitsbeweisen hängt seinerseits mit einem anderen Mangel zusammen. Um zu wissen, was ein Mechanismus lernen kann, muß man erst wissen, was er generieren kann. Dies erfordert einige Erläuterung. Der mögliche Wissensbereich einer klassischen Architektur wird durch eine begrenzte Menge von Regeln oder Operationen, das Programm, bestimmt. Ich habe zuvor Chomskys Beweis dafür erwähnt, daß ein endlicher Automat keine natürliche Sprache repräsentieren kann. Man kann einen solchen Automaten nicht so programmieren, daß er jeden Satz des Deutschen produziert, ohne jemals eine ungrammatikalische Folge von Wörtern hervorzubringen. Wenn ein Wissensbereich wie eine Sprache nicht präsentiert werden kann, dann kann er natürlich erst recht nicht gelernt werden. Aber das Umgekehrte gilt merkwürdigerweise nicht. Wenn ein Automat einen bestimmten Wissensbereich im Prinzip repräsentieren kann, braucht er noch lange nicht in der Lage zu sein, diesen Wissensbereich auch zu lernen[2].

Welchen Wissensbereich kann ein konnektionistisches Netzwerk repräsentieren? Auf diese Frage kann man erst einmal eine einfache Antwort geben. Ein konnektionistisches Netzwerk ist ein endlicher Automat, denn es besteht aus einer endlichen Anzahl Knoten, von denen sich jeder in einer endlichen Anzahl von Zuständen befinden kann. Das Netzwerk als Ganzes kann sich also nur in einer endlichen Anzahl von unterscheidbaren Zuständen befinden. Als endlicher Automat kann ein solches Netzwerk nur re-

2 An dieser Stelle machen es sich die Konnektionisten schwerer als nötig. Ihre erste Frage ist, ob ein Wissensbereich in einem Netzwerk repräsentiert werden kann. Aber wenn etwas nicht erlernbar ist, kann es trotzdem noch repräsentierbar sein.

guläre Mengen repräsentieren (siehe den Text in Abb. 1). Er ist daher grundsätzlich ausgeschlossen, daß komplexere Mengen als reguläre Mengen, z.B. natürliche Zahlen oder die Prädikatenlogik, in einem konnektionistischen Modell repräsentiert werden können.

Hiermit scheint der Vorhang gefallen zu sein, aber so einfach ist es nicht. Zunächst braucht der Aktivationszustand eines Knotens keine diskrete Variable (mit einer endlichen Anzahl von Werten) zu sein. Vor kurzem haben Hornik, Stinchcombe & White (1989) bewiesen, daß Netzwerke mit verborgenen Knoten und einer kontinuierlichen Aktivationsvariablen in der Lage sind, jede meßbare Funktion zu simulieren. Das ist unzweifelhaft ein wichtiges Ergebnis. Es ist das erste Mal, daß hinsichtlich des generativen Vermögens von Netzwerken etwas bewiesen, und nicht nur behauptet wird. Der einflußreiche Konnektionist Elman (1989) folgerte aus diesem Ergebnis sofort, daß konnektionistische Netzwerke *«are effectively Turing machines»*. Das heißt, sie sollten in der Lage sein, jede symbolische Berechnung durchzuführen. Levelt (1990b) zeigte allerdings, daß diese Schlußfolgerung nicht aus dem Resultat von Hornik et al. folgt.

Aber auch, wenn man sich auf Netzwerke mit diskreten Zustandsvariablen beschränkt, könnte man als Konnektionist folgendermaßen antworten: «Die Repräsentation einer komplexen Menge erfordert auch bei einer klassischen Architektur ein unbegrenztes Gedächtnis, das heißt, ein unbegrenztes langes Band in der Turing-Maschine. Die Annahme, daß Menschen so ein unendlich langes Band in ihrem Kopf haben, kann nichts anderes sein als eine theoretische Idealisierung; das muß uns auch gegönnt werden. Mit anderen Worten, wir wollen die Möglichkeit haben, Netzwerke sich unbegrenzt ausdehnen zu lassen».

Die Reaktion ist fair, aber jetzt entstehen wieder neue Schwierigkeiten für den Konnektionismus. Es ist natürlich nicht sinnvoll, von einem unendlich großen Netzwerk auszugehen; dafür benötigt man nämlich eine unendliche Anzahl von Gleichungen. Was man braucht, ist ein Verfahren, das das Netzwerk stets genau so viel vergrößert, wie nötig ist, um eine bestimmte Berechnung durchzuführen. Wenn das Netzwerk z.B. gerade zu klein ist, Sätze wie «Wenn Hans sagt, daß Peter sagt, daß es regnet, schwindelt er» zu erkennen, muß es sich so weit vergrößern können, daß auch das wieder geht.

Aber hier rächt sich, was jubelnd als revolutionärer Erfolg eingebracht worden ist; die Aufhebung der Trennung zwischen Programm und Daten. Wenn ein Netzwerk etwas gelernt hat, und man fügt ein Stück hinzu, dann kann das Erlernte wieder verloren gehen. Mit anderen Worten, eine Vergrößerung des Netzwerks verändert nicht nur die Größe des «Arbeitsgedächtnisses» und den Umfang der Ressourcen, sondern auch die computationale Architektur – die gelernten Regeln[3]. Das passiert einem mit einer klassischen Architektur nicht. Das bringt mich zu dem folgenden Punkt. Konnektionistische Netzwerke sind intolerant gegenüber Wissenserweiterung. Wenn ein konnektionistisches Netzwerk die Menge X gelernt hat, und man lehrt es dann die Menge Y, dann hat es die Menge X wieder vergessen. Dies ist die direkte Folge davon, daß man das Turing-Prinzip – die Trennung von Programm und Daten – aufgegeben hat. Die einzige Art, wie ein Netzwerk etwas dazulernen kann, ist alles Alte wieder von neuem mitzulernen – wahrhaftig ein vortreffliches Modell des menschlichen Lernvermögens!

Ebenso hoffnungslos ist es, eine konnektionistische Erklärung für das *direkte* Erlernen von Regeln, einer unserer wichtigsten Formen der Wissenserweiterung, zu geben. Wenn die Telefonnummern in meiner Stadt ab morgen eine neue Dezimalstelle bekommen, z.B. die Anfangsziffer 2, und ich höre von dieser Regel, dann kann ich diese unmittelbar anwenden, ohne alle Nummer/Namen-Assoziationen neu zu lernen bzw. in einem neuen Telefonbuch aufzusuchen. Ein konnektionistisches Netzwerk muß ganz von neuem trainiert werden. Ein Großteil unseres Handlungswissens besteht aber aus solchen einmalig gelernten Regeln (siehe Levelt, 1990a für mehr Informationen über das hier angesprochene Problem).

Konnektionistische Modelle sind indifferent im Hinblick auf semantische Kohärenz. Dies ist in jeder Hinsicht der Fall. Wenn ein Netzwerk ge-

3 Kürzlich stellte Ash (1989) in einem unveröffentlichten Artikel eine Methode vor, wie man ein Netzwerk langsam wachsen lassen kann, ohne das Erlernte wieder zu verlieren. In einer begrenzten Anzahl von Computersimulationen scheint das auch zu gelingen. Es ist allerdings nichts über die Generalisierbarkeit dieses Resultats bekannt.

lernt hat, daß der Satz «Hans fährt Fahrrad und Peter läuft» wahr ist, weiß es nicht, daß «Hans fährt Fahrrad» wahr ist, oder daß «Peter läuft» wahr ist. Wenn es beide Inferenzen gelernt hat, weiß es danach nicht, daß aus «Peter fährt Fahrrad» und «Hans läuft» folgt: «Peter fährt Fahrrad und Hans läuft». Das kommt daher, daß Regeln und Daten nicht getrennt sind. Wenn die abstrakte Regel, nämlich daß aus P&Q sowohl P auch Q folgen, nach langem Training mit sehr viel Sätzen schließlich doch einigermaßen gelernt ist, weiß das Netzwerk noch immer nicht, daß aus «Hans fährt Fahrrad, und Peter läuft, und Marie arbeitet» folgt, daß Hans Fahrrad fährt. Was aus P&Q&R folgt, muß wiederum völlig neu gelernt werden. Das ist natürlich die Folge davon, daß die Konstituentenstruktur von Aussagen nicht beachtet wird. Es fehlt die Grundlage dafür, aus einer gleichartigen syntaktischen Struktur gleichartige semantische Interferenzen zu ziehen. Viel schlimmer noch: Zum gleichen Preis kann man dem Netzwerk beibringen, aus «Hans fährt Fahrrad» und «Peter läuft» und «Marie arbeitet» abzuleiten, daß Hans nicht Fahrrad fährt. Dasselbe Netzwerk schließt dann aus dem zweigliedrigen Satz, daß Hans wohl Fahrrad fährt, und aus dem dreigliedrigen Satz, daß Hans nicht Fahrrad fährt.

Diese Art von Beispielen läßt sich *ad libitum* fortsetzen. Sie zeigen, daß der Konnektionismus gerade da neutral ist, wo er es nicht sein sollte, als *model of mind.* Er ist in bezug auf semantische Kohärenz neutral; es gibt nichts, was semantische Anarchie verbietet. Es müßte doch die erste Aufgabe eines solchen Modells sein zu erklären, warum der menschliche Geist semantisch nicht willkürlich funktioniert. Das ist genau das, was die klassische Architektur dank ihres Kompositionalitätsprinzips und ihrer Trennung von Programmen und Daten tut (siehe Fodor & Pylyshyn, 1988 für eine ausführliche Diskussion dieser Frage).

Es gibt eine Vielzahl von anderen theoretischen Problemen, die ich hier nur andeuten kann, zum Beispiel die Unmöglichkeit, *types* und *tokens* auseinanderzuhalten (Prince & Pinker, 1988), das unüberwindbare Problem, Variablen zu binden, so daß es keine systematische Basis gibt, um die Referenz von «sich» in «Hans betrinkt sich» und in «Der Vater von Hans betrinkt sich» korrekt zu interpretieren (siehe Abb. 2; wie man sich die menschliche Kognitition ohne die Möglichkeit, Variablen zu binden, vorstellt, ist mir ein Rätsel), die Unmöglichkeit, auf systematische Weise die Logik verschiedener propositionaler Einstellungen zu unterscheiden (man kann nicht gleichzeitig zwei sich gegenseitig ausschließende Dinge glauben, wohl aber wünschen), oder das Fehlen eines Mechanismus für Aufmerksamkeitskontrolle.

Ich will hier noch eine empirische Frage nennen, die mir sehr am Herzen liegt. Wie schon erwähnt, besteht die empirische konnektionistische Forschung darin, daß man ausprobiert, ob ein Bereich X vom Netzwerk erlernt werden kann. Wenn die Antwort einigermaßen positiv ist, bleibt noch die zentrale Frage, ob Menschen X auf dieselbe Art und Weise lernen. Konnektionisten stellen sich diese Frage meistens nicht. Aber in den wenigen Fällen, wo dieser Frage nachgegangen worden ist (durch Forscher außerhalb des konnektionistischen Lagers), war die Antwort für das konnektionistische Modell unerfreulich. Der am besten ausgearbeitete Fall ist der der Vergangenheitsformen. Hier scheint das konnektionistische Modell, das Rumelhart & McClelland (1986) vorgeschlagen haben, wirklich hoffnungslos falsch zu liegen. Das Modell lernte nicht nur alles verkehrt, der Lernprozeß verlief auch noch wesentlich anders als bei Kindern (Pinker & Prince, 1988). Ein essentieller Punkt ist zum Beispiel, daß ein Kind sein Verhalten nach dem Modell erst dann verändert, wenn es mit anderen (neuen) Eingabe-Kontingenzen konfrontiert wird. Dies ist nachweisbar falsch. Allgemeiner gesagt: konnektionistisches Lernen ist strikt frequenzabhängig. Es ist inzwischen aus der empirischen Forschung hinreichend bekannt, daß dies für menschliches Lernen nicht gilt.

Und nun?

Angesichts der Tatsache, daß ein konnektionistisches Modell des menschlichen Geistes nicht in der Lage ist, die wichtigsten Kennzeichen des kognitiven Funktionierens zu behandeln, stellt sich nun die Frage: Wofür können konnektionistische Netzwerke dann doch nützlich sein? Auf diese Frage habe ich zwei Antworten, eine pragmatische und eine prinzipiellere. Die pragmatische Reaktion ist wie folgt: Laß jeden tun, was

er will. Es gibt bestimmt begrenzte Probleme, die sich mit einem konnektionistischen Modell bearbeiten lassen, wie etwa statistische Inferenz, das Verschärfen von unscharfen Fotographien und *content-addressable storage* von festen Datenbeständen (wo übrigens die klassischen Lösungen noch immer viel besser sind). Die Wissenschaft ist eine Art anarchistischer Markt, und wir werden schon merken, welches Produkt seinen Wert behält. Diese Reaktion ist allerdings nicht ganz befriedigend. Wir werden schließlich dafür bezahlt, gut nachzudenken, deswegen hier eine prinzipiellere Reaktion. Welcher theoretische Platz kann den konnektionistischen Netzwerken in einer Theorie der menschlichen Kognition eingeräumt werden? Meiner Meinung nach kann es kein anderer sein als der eines potentiellen Implementierungsmediums. (Ich schließe mich hier der Auffassung von Fodor & Pylyshyn [1988] an). Es ist verhältnismäßig, aber nicht vollkommen irrelevant, wie ein kognitives Modell implementiert wird. So sind viele Beschwerden, die Konnektionisten angesichts der klassischen Architektur äußern, zum Beispiel, daß diese nur für langsame serielle Verarbeitung zu gebrauchen sei und keine Toleranz gegenüber Rauschen habe etc., in Wirklichkeit Beschwerden gegen ihre Implementierung in Von-Neumann-Maschinen. Die meisten kognitiven Modelle, etwa die des Sprachverstehens und der Sprachproduktion, haben eine vollkommen parallele Architektur (Levelt, 1989), und lassen sich deshalb nur mühsam in Von-Neumann-Maschinen implementieren. Es ist sicher denkbar, daß sich einige Aspekte solcher Architekturen besser in einem konnektionistischen Netzwerk implementieren lassen, und es ist die Mühe wert, dies auszuprobieren.

Ich finde es übrigens selbst hier etwas verfrüht, hohe Erwartungen zu wecken. Was nämlich in einer verteilten Netzwerkrepräsentation so gut wie unmöglich ist, ist *multiple tasking*, die Fähigkeit, ein und dasselbe Netzwerk parallel mehrere Aufgaben gleichzeitig ausführen zu lassen. Die Konnektionisten sprechen andauernd von «massiver paralleler Verarbeitung», aber wenn ein Netzwerk auf eine Aufgabe aus dem einen Wissensbereich programmiert ist, dann geht dieses Wissen, wie wir sahen, verloren, wenn es lernen muß, eine andere Aufgabe aus einem anderen Wissensbereich auszuführen. Die einzige Lösung besteht demnach darin, jede Fähigkeit und jeden Teilprozeß in einem eigenen kleinen Netzwerk unterzubringen. Die kleinen Netzwerke können dann unabhängig voneinander und parallel arbeiten. Eine Schwierigkeit dabei ist dann wieder, daß das Wissen, das im Netzwerk X gespeichert ist, nicht auf Netzwerk Y übertragbar ist. Das eine Netzwerk kann das Wissen in einem anderen Netzwerk nicht «lesen». So entstehen enorme Informationsübertragungs-Probleme, selbst in so einem modular aufgebauten System von Netzwerken.

Es gibt auch noch andere Gründe anzuzweifeln, daß solche Netzwerke für die Implementierung brauchbar sind. Ein geflügeltes Wort der Konnektionisten lautet *brain-style modelling*. Konnektionistische Netzwerke sollten neuronalen Netzwerken gleichen, also dem cerebralen Substrat, in dem alle Kognitionen letztlich implementiert sind. Wir wissen inzwischen, daß das nicht im entferntesten der Fall ist. (Siehe z.B., was der konnektionistische Vordenker Smolensky [1988] darüber sagt. Siehe auch, wie Crick [1989] den neuronalen Ansprüchen den Boden unter den Füßen wegzieht – vor allem der neurologischen Realität der *backward propagation*, einem für den Konnektionismus essentiellen Begriff. Der Begriff steht im Widerspruch zu dem Einbahnstraßenverkehr in der Signalübertragung von Neuronen). Ein bißchen mehr *brain-style modelling* könnte sicher nicht schaden. Warum lassen die Konnektionisten als einzige Operation in ihren Netzwerken die Addition zu, das lineare oder nicht-lineare Summieren von Aktivationsströmen? Echte Neuronen sind zu viel mehr instande, vor allem zu allerlei logischen Verschaltungen. Man könnte die logischen Verschaltungen vermutlich wohl allesamt in einem konnektionistischen Netzwerk programmieren, aber das wäre lediglich eine durch den Formalismus verursachte Komplikation. Die Implementierung auch nur einigermaßen komplexen Verhaltens erfordert logische Verschaltung. Aber solche Verschaltungen sind für die Konnektionisten tabu.

Schlußfolgerung

Die Verfügbarkeit von sehr großen Rechnern hat eine lange intellektuelle Tradition im westlichen Denken zu neuem Leben erweckt, den Assozia-

tionismus. Aber die heutige Reinkarnation, der Konnektionismus, unterliegt der gleichen Kritik, der frühere Varianten wie die von Hume und auch der Behavioristen ausgesetzt waren. Als Modell des menschlichen Geistes ist der Konnektionismus keine Alternative für das, was ich die klassische computationale Architektur genannt habe. Das Beste, was wir erhoffen können, ist, daß das neue Spielzeug sich zumindest zum Modellieren einiger Teilaspekte der menschlichen Kognition eignet, vor allem solcher Aspekte, bei denen der Wissensbestand begrenzt und nicht rekursiv ist. Außerdem lassen sich konnektionistische Netzwerke vielleicht als mehr oder weniger geeignete Implementierungsmedien für eine Anzahl kognitiver Operationen gebrauchen. Aber mehr sollte man dann auch nicht davon erwarten.

Literatur

Chomsky, M. (1957). *Syntactic Structures.* Den Haag: Mouton.

Crick, F. (1989). The recent excitement about neural networks. *Nature, 337,* 129–132.

Elman, J. L. (1989). *Representation and Structure in Connectionist Models.* CRL Technical Report 8903.

Fodor, J. A. & Pylyshyn, Z. W. (1988). Connectionism and cognitive architecture: A critical analysis. *Cognition, 28,* 3–71.

Goebel, R. (1990). A connectionist approach to high-level cognitive modeling. *Proceedings of the Twelfth Annual Conference of the Cognitive Science Society,* Cambridge, Ma.

Hebb, D. O. (1949). *The organization of behavior.* New York: Wiley.

Hornick, K., Stinchcombe, M. & White, H. (1989). *Multilayer feedforward networks are universal approximators.* Discussion Paper 88-45R. Department of Economics, UCSD.

Hunter, L. E. (1988). Some memory, but no mind. *Behavioral and Brain Sciences, 11,* 37–38.

Laird, J. E., Rosenbloom, P. S. & Newell, A. (1986). Chunking on Soars: The anatomy of a general learning mechanism. *Machine Learning, 1,* 11–46.

Levelt, W. J. M. (1974). *Formal grammars in linguistics and psycholinguistics.* (3 Vols). Den Haag: Mouton.

Levelt, W. J. M. (1988). Onder sociale wetenschappen. Toegelicht aan psychologie, economie en taalkunde. *Mededelingen KNAW,* Deel 51, no 2.

Levelt, W. J. M. (1989). *Speaking: From Intention to Articulation.* Cambridge, Mass.: MIT Press.

Levelt, W. J. M. (1990a). Are multilayer feedforward networks effectively Turing machines? *Psychological Research, 52,* 153–157.

Levelt, W. J. M. (1990b). On learnability, empirical foundations, and naturalness. Commentary on S. J. Hanson and J. Burr, What connectionist models learn: Learning and representation in connectionist networks. *Behavioral and Brain Sciences, 13,* 501.

McClelland, J. L. & Rumelhart, D. E. (1981). An interactive activation model of context effects in letter perception: Part 1. An account of basic findings. *Psychological Review, 88,* 375–407.

McCulloch, W. S. & Pitts, W. (1943). A logical calculus of the ideas immanent in neural nets. *Bulletin of Mathematical Biophysics, 5,* 115–137.

Minsky, M. L. & Papert, S. A. (1969). *Perceptrons.* Cambridge, Mass.: MIT Press.

Pinker, S. & Prince, A. (1988). On language and connectionism: Analysis of a parallel distributed processing model of language acquisition. *Cognition, 28,* 73–193.

Prince, A. & Pinker, A. (1988). Subsymbols aren't much good outside of a symbol-processing architecture. *Behavioral and Brain Sciences, 11,* 46–47.

Rosenblatt, D. E. (1962). *Principles of neurodynamics.* New York: Spartan.

Rumelhart, D. E. & McClelland, J. L. (Eds.) (1986). *Parallel distributed processing.* Vol. I. Cambridge, Mass.: MIT Press.

Smolensky, P. (1988). On the proper treatment of connectionism. *Behavioral and Brain Sciences, 11,* 1–42.

Turing, A. M. (1936). On computable numbers with an application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society, 42,* 230–265.

Prof. Dr. Willem Levelt, Max-Planck-Institut für Psycholinguistik, Wundtlaan 1, NL-6525 XD Nijmegen